REMARKS

Claims 1-21, 23-26, and 28-42 remain pending in the present application. Claims 7, 17, 20, and 42 have been amended to correct errors in antecedent basis. Claims 21 and 26 have been amended to include the limitations of claims 22 and 27 respectively. Claims 22 and 27 have been cancelled. claims 24 and 29 have been amended to correct claim dependency.

Claim Objections

Claims 7, 17, 20, and 42 were objected to as lacking antecedent basis. Each of these claims have been amended to correct these errors.

Claim Rejections under 35 U.S.C. § 103(a)

Claims 1, 3-9, 11-19, 21-31, 33-35, and 37-41 were rejected were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,630,083 to Carbine ("Carbine") in view of U.S. Patent No. 5,765,220 to Kipp ("Kipp"). Claims 10, 20, 36 and 42 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Carbine and Kipp in view of "common art."

As described in the Background section of the present application, programmable logic arrays ("PLAs") provide micro-operation information for short instruction flows of usually four or less micro-operations while a micro-operation read only memory ("UROM") is used to provide micro-operation information for long instruction flows of more than four micro-operations. Processor efficiency is impacted on long instructions because the processor has to switch to the slower UROM to access the long instructions.

According to embodiments of the present invention a micro-operation (Uop) cache (e.g., element 160 in Fig. 1) is provided in addition to a micro-operation memory (e.g., a UROM; element 170). Thus, a trace pipe is provided that receives micro-operation information from both the micro-operation memory and the micro-operation cache. In one embodiment, the micro-operation memory stores all micro-operation information and the micro-operation cache stores micro-operation information for the most commonly used instructions and/or micro-operations that are read from the micro-operation memory. A potential advantage of this configuration is that all micro-operation information would come from a single structure, instead of receiving information for short instructions from programmable logic arrays ("PLAs") and receiving information for long instructions from a UROM.

Claim 1, for example refers to a micro-operation cache that is coupled to an instruction pointer sequencing logic/predictor component, and a micro-operation memory coupled to the micro-operation cache. Claim 31 includes similar limitations. Claim 11 refers to a micro-operation cache to store a plurality of frequently used micro-instruction operations and a micro-operation memory to store a plurality of micro-instruction operations. Claim 27 includes similar limitations. Claims 21 and 26 refer to storing in a cache a plurality of commonly used micro-operations for quick access and storing a plurality of micro-operations including the plurality of commonly used micro-operations. The cited references, taken singularly or in combination fail to teach or suggest these features.

Carbine refers to a decoder for decoding multiple instructions in parallel. The only cache in Carbine is the instruction cache 804 shown in Fig. 8. Such a cache supplies macroinstructions to be decoded for execution. If the required macroinstruction is not found in the instruction cache, then the instruction is fetched from main memory 800 (see Col. 16, line 60 to Col. 17, line

6). Accordingly, with respect to the claims, which recite a micro-operation cache, no such cache is shown in Carbine. Cache 804 stores macroinstructions that are to be decoded into micro-operations (see Col. 2, lines 11-14). Though the Office Action states that such is shown in Carbine, none of the cited sections of Carbine refers to a cache of any kind.

The Office Action states that Carbine fails to "disclose the use of an instruction pointer queue, rather than an instruction queue itself." Though Kipp is used to describe such an instruction pointer queue, Kipp fails to make up for the deficiencies of Carbine. Kipp refers to reducing instruction address storage in a super-scalar processor. In Kipp, when a branch is mispredicted, it is necessary to read address of instructions to be executed. Since the addresses are stored with the instructions themselves in an instruction queue, the entire instruction must be fetched with the instruction to read the address. Such takes up space, increases power consumption, and slows the processor. To address this, Kipp provides a cache to store the instruction addresses alone. The Office Action states that Carbine "would be motivated to utilize this method to minimize queue storage space (and, likely, power) as described in Kipp col 5, lines 50-55...." Even assuming that Kipp provides an instruction pointer queue, the combination of Kipp and Carbine fail to teach or describe the micro-operation cache and micro-operation memory recited in each of the pending claims.

Since features of the claims are neither taught nor suggested by Carbine and/or Kipp, reconsideration and withdrawal of the rejection of claims 1, 3-21, 23-26, 28-31, and 33-41 under 35 U.S.C. § 103(a) is respectfully requested.

CONCLUSION

For all the above reasons, the Applicant respectfully submits that this application is in condition for allowance. A Notice of Allowance is earnestly solicited.

The Examiner is invited to contact the undersigned at (202) 220-4255 to discuss any matter concerning this application. The Office is hereby authorized to charge any additional fees or credit any overpayments under 37 C.F.R. § 1.16 or § 1.17 to Deposit Account No. 11-0600.

Respectfully submitted,
KENYON & KENYON LLP

Dated: October 18, 2006

By: _____
Shawn W. O'Dowd
Reg. No. 34,687

KENYON & KENYON LLP
1500 K Street, NW
Suite 700
Washington DC, 20005
(202) 220-4200 telephone
(202) 220-4201 facsimile
DC1-632603